UNITED STATES UTILITY PATENT APPLICATION

SYSTEM AND METHOD FOR DETECTING PROCESS AND

NETWORK FAILURES IN A DISTRIBUTED SYSTEM

INVENTOR:

Roger A. FLEMING
2411 Coventry Court
Fort Collins, CO 80526

SYSTEM AND METHOD FOR DETECTING PROCESS AND NETWORK FAILURES IN A DISTRIBUTED SYSTEM

The following applications containing related subject matter and filed concurrently with the present application are hereby incorporated by reference: Serial No. TBD, Attorney Docket No. 10010269-1, entitled System and Method for Detecting Process and Non-Process Failures in a Distributed System Having Multiple Independent Networks; Serial No. TBD, Attorney Docket No.10010271-1, entitled Probationary Members; and Serial No. TBD, Attorney Docket No. 10010270-1 and entitled Adaptive Heartbeats.

FIELD OF THE INVENTION

The present invention is generally related to monitoring computer processes in a distributed system. More particularly, the present invention is related to detecting process and network failures in a distributed system.

BACKGROUND OF THE INVENTION

In recent years, reliable, high performance computer systems have been, and still are, in great demand. Users have also demanded the introduction and propagation of multi-processor distributed computer systems to support their computing processes (e.g. simulations, parallel processing, etc.). A distributed computer system generally includes a collection of processes and a collection of execution platforms (i.e., hosts). Each process may be capable of executing on a different host, and collectively, the processes function to provide a computer service. A failure of a critical process in a distributed system may result in the service halting. Therefore, techniques have been implemented for detecting a failure of a process in a timely manner, such that an appropriate action can be taken.

A conventional technique for detecting failure of a process includes the use of heartbeats, which are messages sent between processes at regular intervals of time. According to the heartbeat technique, if a process does not receive a heartbeat from a remote process prior to the expiration of a predetermined length of time, i.e., the heartbeat timeout, the remote process is suspected to have failed. Corrective action, such as eliminating the suspected process, may thus be taken.

A remote process not transmitting a heartbeat may not be an indication of a failure in the remote process. Instead, a network failure may have prevented a process from receiving a heartbeat from the remote process, especially when multiple processes in a distributed system are communicating over a common network. For example, a network failure may include a

5   network pause (i.e., a temporary condition that prevents communication on a network) or a less temporary network failure, such as a hardware failure for hardware facilitating transmission on the network. A network pause, for example, can be the result of heavy, high-priority traffic over a network link, sometimes caused by other processes (e.g., remote machine backups). If the network pause endures for a period of time greater than the

10   heartbeat timeout or if a network failure occurs, each process waiting for a heartbeat transmitted over the network in the distributed system may suspect the other processes of failing. Then, each process may take unnecessary corrective actions, such as eliminating and/or replacing the suspected processes from the distributed system, which can cause each service provided by the processes in the distributed system to be halted. If network

15   conditions can be detected, appropriate corrective action could be taken, such as establishing connections between the distributed system processes using alternative paths.

SUMMARY OF THE INVENTION

An aspect of the present invention is to provide a system and method for detecting

20   and distinguishing between a process failure and a network failure in a distributed system.

In one respect, the present invention includes a system and method for detecting a process failure in a distributed system. A process in the distributed system is connected to a plurality of other processes in the distributed system via a network. If the difference in the period of time to receive a heartbeat from a first of the plurality of processes and a period of

25   time to receive a heartbeat from a second process of the plurality of processes exceeds a process failure threshold, the second process is suspected of failing.

In another respect, the present invention includes a system and method for detecting a network failure in the distributed system. A process in the distributed system monitors a plurality of other processes in the distributed system via a network. If the process fails to

30   receive a heartbeat from any one of the plurality of processes within a network failure time limit, the network in the distributed system is suspected of failing.

The methods of the present invention include steps that may be performed by computer-executable instructions recorded on a computer-readable medium.

The present invention provides low cost simplistic techniques for detecting network and process failures in a distributed system. Accordingly, corrective action may be taken when failures are detected. Therefore, down-time for a service provided by the processes in the distributed system may be minimized. Those skilled in the art will appreciate these and other advantages and benefits of various embodiments of the invention upon reading the following detailed description of a preferred embodiment with reference to the below-listed drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the accompanying figures in which like numeral references refer to like elements, and wherein:

Fig. 1 illustrates an exemplary block diagram of a distributed system employing the principles of the present invention;

Fig. 2 illustrates a flow-diagram of an exemplary embodiment of a method employing the principles of the present invention;

Fig. 3 illustrates a flow-diagram of another exemplary embodiment of a method employing the principles of the present invention; and

Fig. 4 illustrates a flow-diagram of another exemplary embodiment of a method employing the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one of ordinary skill in the art that these specific details need not be used to practice the present invention. In other instances, well known structures, interfaces, and processes have not been shown in detail in order not to unnecessarily obscure the present invention.

Fig. 1 shows a distributed system 100 employing the principles of the present invention. The distributed system 100 includes host 1, host 2 and host 3 executing process A, process B and process C, respectively. Processes A-C function to provide a service to a plurality of users via distributed system 100. Hosts 1-3 are connected via bi-directional communication paths 110, 120 and 130. Communication paths 110, 120 and 130 include

network links in one network 150. Hosts 1-3 are typical nodes in a distributed system and can include a data processing system, memory and network interface, all of which are not specifically shown. It will be apparent to those of ordinary skill in the art that an arbitrary number of hosts in distributed system 100 may be supported in an arbitrary configuration.

5   Furthermore, each host may execute one or more processes.

An administration function performed by distributed system 100 can include detecting failure of one or more of processes A-C, such that corrective action (e.g., eliminating and/or replacing a failed process) can be taken when a process fails. For example, a failed process may be removed from a "view", when a consensus is reached that the process has failed.

10   Accordingly, processes A-C, executing on hosts 1-3 respectively, transmit heartbeats on communication paths 110-130 in network 150 to detect a process failure. Process A may utilize a process failure algorithm for detecting a failure of a process in system 100. The process failure algorithm includes comparing the difference between a period of time to receive a heartbeat from a first process and period of time to receive a heartbeat from a

15   second process to a process failure threshold. For example, process A monitors processes B and C by monitoring heartbeats transmitted on communication paths 110 and 130 from processes B and C, respectively. If the difference between a period of time to receive a heartbeat from process B and a period of time to receive a heartbeat from process C exceeds a process failure threshold, process B is suspected of failing. The process failure threshold may

20   be a predetermined threshold or a threshold that can automatically adapt to varying network conditions. It will be apparent to one of ordinary skill in the art that the threshold may be determined based upon the network configuration, average network traffic and/or other factors relevant to network transmission.

System 100 may also detect failure of network 150 using a network failure algorithm,

25   such as determining whether a heartbeat is received from any process in system 100 prior to expiration of a network failure time limit. For example, process A monitors processes B and C by monitoring heartbeats transmitted on communication paths 110 and 130 from processes B and C, respectively. If process A fails to receive a heartbeat from any one of processes B and C within a network failure time limit, network 150 is suspected of failing. Similarly to

30   the process failure threshold, the network failure time limit may be predetermined or adaptive. It will be apparent to one of ordinary skill in the art that the time limit may be determined based upon the network configuration, average network traffic and other factors relevant to network transmission.

A network failure may include a network condition that prevents communication on the network for a predetermined period of time. For example, a network failure may include a network pause (i.e., a temporary condition that prevents communication on a network) or a less temporary network failure, such as a hardware failure for hardware facilitating

5   transmission on the network. A network pause, for example, can be the result of heavy, high priority traffic over a network link, sometimes caused by other processes (e.g., remote machine backups).

Based on the monitoring of processes B and C, process A may take appropriate corrective action. For example, when process A determines that process B has failed, process

10   A can eliminate and/or replace process B. Alternatively, when process A determines that a network failure may have occurred, process A may take a different action, such as waiting for a condition causing a network pause to clear or attempting to establish new communication path(s) over a different network or alternative paths within network 150.

A flow-diagram, shown in Fig. 2, illustrates an exemplary embodiment of a method

15   200 for implementing the network failure algorithm of the present invention. The steps shown in Fig. 2 are described with respect to processes A-C in distributed system 100. It will be apparent to one of ordinary skill in the art, however, that the method shown in Fig. 2 is applicable to distributed systems having a variety of configurations and having a process monitoring more than two processes.

20   In step 210, process A determines whether a heartbeat is received from any process (e.g., process B or process C) in network 150 prior to the expiration of the network failure time limit. If a heartbeat is not received prior to the expiration of the network failure time limit, network 150 is suspected to have failed and appropriate corrective action may be taken (step 215). If a heartbeat is received prior to the expiration of the network failure time limit,

25   the network failure time limit is reset (step 220). Then, method 200 is repeated.

A flow-diagram, shown in Fig. 3, illustrates an exemplary embodiment of a method 300 including the steps of the process failure algorithm of the present invention. The steps shown in Fig. 3 are described with respect to processes A-C in distributed system 100. It will be apparent to one of ordinary skill in the art that the method shown in Fig. 3 is applicable to

30   distributed systems having a variety of configurations and having a process monitoring more than two processes. Also, it will be apparent to one of ordinary skill in the art that the process failure algorithms may be implemented using a plurality of techniques.

In step 305, a first period of time between an instance a last heartbeat was received from a first process (e.g., process B) and a later instance in time is measured. In step 310, a second period of time between an instance a last heartbeat was received from a second process (e.g., process C) and the later instance in time is measured. In step 320, the difference between the first and second periods of time is calculated. In step 330, the difference is compared to the process failure threshold. If the difference exceeds the process failure threshold, the second process is suspected of failing (step 340), and appropriate corrective action may be taken. If the difference does not exceed the process failure threshold, a failure of the second process is not suspected (step 350).

A flow-diagram, shown in Fig. 4, illustrates an exemplary embodiment of a method 400 implementing the process failure algorithm of the present invention in a distributed system. The steps shown in Fig. 4 are described with respect to processes A-C in distributed system 100. It will be apparent to one of ordinary skill in the art, however, that the method shown in Fig. 4 is applicable to distributed systems having a variety of configurations and having a process monitoring more than two processes.

In step 405, process A receives a heartbeat from a first process (e.g., process B in system 100). In step 410, a timer is started for detecting a heartbeat timeout of a second process (e.g., process C) in distributed system 100 that is monitored by process A. In step 415, process A determines whether a heartbeat is received from process C. If a heartbeat is received from process C, the timer is cancelled (step 420). If a heartbeat is not received from process C, process A determines whether the heartbeat timeout for process C is expired (step 425). The heartbeat timeout may be predetermined or adaptive, similar to the process failure threshold. An adaptive heartbeat timeout technique is described in co-pending U.S. Pat. Application No. TBD, entitled Adaptive Heartbeats and incorporated by reference herein. It will be apparent to one of ordinary skill in the art that a predetermined heartbeat timeout may be determined based upon the network configuration, average network traffic and other factors relevant to network transmission.

If the heartbeat timeout is expired, process A suspects a failure of process C (step 430), and process A may take appropriate corrective action. If the heartbeat timeout is not expired, process A determines whether a heartbeat is received from another process (step 415).

The methods shown in Figs. 2-4 detect process and network failures. Accordingly, corrective actions tailored to the type of failure detected can be taken to reach a timely

solution. Thus, down-time is limited for service(s) facilitated by processes executing in a distributed system.

The methods shown in Figs 2-4 and described above can be performed by a computer program. The computer program can exist in a variety of forms both active and inactive. For example, the computer program can exist as software possessing program instructions or statements in source code, object code, executable code or other formats; firmware program(s); or hardware description language (HDL) files. Any of the above can be embodied on a computer readable medium, which include storage devices and signals, in compressed or uncompressed form. Exemplary computer readable storage devices include conventional computer system RAM (random access memory), ROM (read only memory), EPROM (erasable, programmable ROM), EEPROM (electrically erasable, programmable ROM), and magnetic or optical disks or tapes. Exemplary computer readable signals, whether modulated using a carrier or not, are signals that a computer system hosting or running the computer program can be configured to access, including signals downloaded through the Internet or other networks. Concrete examples of the foregoing include distribution of executable software program(s) of the computer program on a CD ROM or via Internet download. In a sense, the Internet itself, as an abstract entity, is a computer readable medium. The same is true of computer networks in general.

Also, the methods shown in Figs 2-4 and described above may be performed by a process facilitating a service, such as process A in distributed system 100, or performed by a separate process executed on a host in a distributed system.

While this invention has been described in conjunction with the specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. There are changes that may be made without departing from the spirit and scope of the invention.